**ТЕХНИЧЕСКОЕ ЗАДАНИЕ № U001**

**на разработку программы  
«****SymphonyNext»**

**Назначение программы:** программа предназначена для запуска КИС «Симфония» в различных конфигурациях, которые формируются под конкретного пользователя.

**Основные функции:**

1. Автоматическое определение конфигурации запуска программы и сохранения её в настроечный файл, в персональном каталоге пользователя («Мои документы»).
2. Запуск программы, формирующей главное окно КИС «Симфония» в соответствии с полученной конфигурацией.
3. Запуск программного модуля, позволяющего пользователю редактировать настроечный файл под собственные нужды.
4. Обновление настроечных файлов запуска КИС «Симфония» из корневого каталога системы при старте программы SymphonyNext

**Требования к программе и основные алгоритмы:**

1. **Среда разработки.** Программа создается средствами Delphi с использованием библиотеки FireMankey с возможностью компилирования проекта под любую, доступную для среды разработки, платформу (Windows, Android, iOS).
2. **Формат и структура настроечного файла.** Настроечный файл представляет собой XML-файл с расширением .scf (**s**ymphony **c**ommand|**c**onfig **f**ile). Нестандартное расширение вводится для того, что бы можно было сформировать ассоциацию этого расширения с разрабатываемой программой и автоматически запускать её, при запуске файла с расширением .scf. Настроечный файл должен иметь следующую структуру:
   1. Строка описывающая формат файла и используемую кодировку:   
      <?xml version="1.0" encoding="UTF-8" ?>.
   2. Корневой XML документ с тегом SCF и атрибутом VERSION, в котором указывается версия настроечного файла. Строка, описывающая версию настроечного файла не должна превышать 64 символов и должна состоять из символов допустимых в именах файловой системы Windows.  
      <SCF VERSION ="строка описывающая версию файла">
   3. Корневой XML-документ должен содержать следующие узлы:
      1. Узел ROOTPATH, содержащий корневой каталог с развернутой КИС «Симфония»:  
         <ROOTPATH>\\symsrv\symphony</ROOTPATH>
      2. Узел BIT, содержащий число 32 или 64, означающее разрядность используемой системы. Число 32 является значением по умолчанию. Все числа, кроме 64, а так же отсутствие какого либо числа, интерпретируются как число 32.  
         <BIT>64</BIT>
      3. Узел DB, содержащий другие узлы, описывающие используемую базу данных:
         1. Узел TYPE, содержащий строку с типом используемой базы данных
         2. Узел SERVER, содержащий строку с именем сервера, на котором развернута база данных. Для серверов с oracle, данный узел должен содержать имя, заданное в файле tnsnames.ora локального компьютера
         3. Узел DATABASE, содержит строку с именем развернутой на сервере базы данных. Для сервера oracle узел DATABASE игнорируется.
      4. Узел AUTORUNLIST содержит список плагинов без расширения .bpl, разделенные запятыми. Плагины указанные в данном списке, будут запущены при старте системы.
3. **Ассоциация с расширением .scf.** При старте программа SymphonyNext пытается создать ассоциацию с расширением файлов .scf, при условии что данную операцию поддерживает операционная среда, под которой запущено приложение.
4. **Анализ параметров командной строки.** Программа должна проанализировать полученные при старте параметры командной строки и продолжить работу в одном из режимов: «Старт без настроечного файла», «Старт с настроечным файлом», «Редактирование настроечного файла» и «Обновление настроечных файлов».
5. **Старт без настроечного файла.** Режим предназначен для первого запуска SymphonyNext или при потере настроечного файла. Основная задача данного режима: сформировать персональный настроечный файл и перейти в режим «Старт с настроечным файлом». Выполнение программы в данном режиме предусматривает следующие шаги:
   1. Поиск в каталоге с запущенной программой SymphonyNext файла default.scf, который будет содержать настройки по умолчанию. Если файл найден, то настройки считываются, после чего происходят дальнейшие проверки.
   2. Определение корневого каталога. Программа последовательно проверяет считанный каталог из файла default.scf (см. 5.1) и каталог из которого запущена программа SymphonyNext на соответствие требованиям к структуре корневого каталога системы. Если эти каталоги не удовлетворяет требованиям, то корневой каталог запрашивается у пользователя. Если корневой каталог определить не удалось, то программа прекращает свою работу. Каталог удовлетворяет требованиям корневого каталога системы, если в нем присутствует папка bin или bin64, в которой находится файл SymMng.exe
   3. Определение разрядности системы. Программа определяет разрядность операционной системы и наличие файла SymMng.exe в папке bin64, для 64-х битных систем, и в папке bin, для 32-х битных систем. Разрядность выбирается исходя из возможностей выполнения нужной версии программы SymMng.exe с учетом указанной в файле default.scf (см. 5.1) значения разрядности. (если разрядность указана).
   4. Определение типа базы данных. Программа в соответствии с выбранной разрядностью (см. 5.3) ищет в папке bin или bin64 файлы SymDB\*.bpl и SymTask\*.bpl. По найденным файлам SymDB\*.bpl определяются возможные типы используемых баз данных. Предпочтение отдается типу базы данных, для которой так же имеется файл SymTask\*.bpl. Если таких типов несколько, то выбирается тип, указанный в файле default.scf (см. 5.1), а если его там нет, то любой из обнаруженных типов.
   5. Полученные в п. 5.2 - 5.4 параметры записываются в файл SymphonyNext.scf в персональную папку пользователя, а на рабочем столе создается ярлык на этот файл.
   6. Программа проверяет наличие обновлений настроечных файлов в корневом каталоге системы и если они обнаружены, то выполняет шаги предусмотренные режимом «Обновление настроечных файлов» (см. 8)
   7. Если п. 5.5 выполнен успешно, то программа переходит в режим Старт с настроечным файлом.
6. **Старт с настроечным файлом.** Режим предназначен для определения конфигурации запуска КИС «Симфония» и её запуска на выполнение. Выполнение программы в данном режиме предусматривает следующие шаги:
   1. Считывание параметров из полученного в командной строке настроечного файла. По полученным параметрам осуществляются следующие проверки:
   2. По пути к корневому каталогу системы и разрядности, программа определяет наличие файла SymMng.exe. Если в настроечном файле указана 64-х битная разрядность и файл в папке bin64 не найден, то программа автоматически понижает разрядность до 32-х бит и повторяет проверку. Если файл SymMng.exe после всех проверок не найден, то программа переходит в режим Старт без настроечного файла (п. 5) с выдачей сообщения пользователю о возникшей проблеме.
   3. По пути к корневому каталогу системы и полученной на предыдущем этапе разрядности системы, программа ищет файл SymDB\*.bpl. Если файл не найден, то программа переходит в режим Старт без настроечного файла (п. 5) с выдачей сообщения пользователю о возникшей проблеме.
   4. Если настроечные данные прошли проверки, то программа проверяет наличие обновлений настроечных файлов в корневом каталоге системы и если они обнаружены, то выполняет шаги предусмотренные режимом «Обновление настроечных файлов» (см. 8)
   5. Программа запускает найденный файл SymMng.exe и прекращает свою работу. В командной строке SymMng.exe передается имя используемого настроечного файла.
7. **Редактирование настроечного файла.** Данный режим предназначен для редактирования настроечного файла пользователем системы. Данный режим запускается, если в командной строке SymphonyNext передано имя нового или существующего настроечного файла, а так же имеется параметр с одним из следующих значений: «-config», «-с», «/config», «/c».   
     
   Программа SymphonyNext ищет в каталоге bin файл scfmng.bpl и если он найден, то данный пакет загружается. В загруженном пакете ищется экспортируемая функция ConfigSymphony:  
     
   **function** ConfigSymphony(**var** ConfigData: **String**): Boolean ;  
     
   Параметр ConfigData содержит XML-данные настроечного файла. Функция возвращает True, если пользователь отредактировал данные и нажал в диалоговом окне кнопку Ok. В данном случае программа SymphonyNext сохраняет полученные через параметр ConfigData настроечные данные в файл.   
     
   Если в командной строке обнаружен параметр с одним из значений: «-update», «-u», «/update», «/u», то программа сохраняет файл инициализирующий обновление локальных настроечных файлов у всех пользователей (см. п. 8). Исправления записываются не в переданный файл, а в новый настроечный файл в папку update/config корневого каталога системы. Имя нового файла должно совпадать с версией старого настроечного файла, причем в новом настроечном файле XML-документ будет иметь атрибут VERSION с новым значением.  
     
   После выполнения редактирования настроечного файла программа SymphonyNext прекращает свою работу.
8. **Обновление настроечных файлов.** Данный режим предназначен для обновления настроечных файлов на локальной машине пользователя в соответствии с новыми настройками, размещенными в корневом каталоге системы. Выполнение программы в данном режиме предусматривает следующие шаги:
   1. В корневом каталоге системы, указанном в загруженном настроечном файле, в папке UPDATE/CONFIG ищется файл обновления конфигурации с расширением .scf и именем совпадающим с версией текущего .scf файла. Если такого файла нет, то обновлений не требуется и выполнение шагов, предусмотренных данным режимов, прекращается.
   2. Файл обновления конфигурации должен содержать новую версию, а так же обновляемые данные. В загруженный конфигурационный файл вписывается новая версия и обновляемые данные, которые проверяются на корректность по п. 6.2 и п. 6.3
   3. В каталоге bin корневого каталога системы ищутся файлы scfupd\*.bpl. Данные пакеты должны экспортировать процедуру ExecuteUpdate. Предполагается, что данные пакеты при выполнении данных процедур будут выполнять работы по дальнейшему обновлению системы. Например, поиск файлов tnsnames.ora на локальных жестких дисках и их обновление. Условиями запуска данных файлов является соответствие шаблону имени пакета, размещение в папке bin и экспортирование процедуры ExecuteUpdate. Алгоритмы работы данной процедуры не регламентируются, но могут быть сформулированы рекомендации для программистов, разрабатывающих пакеты scfupd\*.bpl.

**Возможный сценарий миграции КИС «Симфония» на новые сервера**

КИС «Симфония» расположена в корневом каталоге [\\alpha\symphony\](file:///\\alpha\symphony\) и использует базу данных oracle на сервере omega, для подключения к которой используется имя ORCL, которое определено в локальных файлах tnsnames.ora. Настроечные файлы пользователей содержат версию «sym-1000»

Требуется перенести файлы КИС «Симфония» в корневой каталог [\\betta\Симфония](file:///\\betta\Симфония) , а в качестве базы данных использовать новый сервер баз данных gamma.

Процесс обновления:

1. Создается пакет [\\alpha\symphony\bin\scfupdtnsnm.bpl](file:///\\alpha\symphony\bin\scfupdtnsnm.bpl) (см. 8.3), который экспортирует процедуру ExecuteUpdate. Данная процедура, например:
   1. Считывает из файла [\\alpha\symphony\update\config\tnsnames-sym-1000.ora](file:///\\alpha\symphony\update\config\tnsnames-sym-1000.ora) новые настройки для подключения к базе данных oracle, работающей на новом сервере
   2. Ищет файлы tnsnames.ora на локальных дисках системы. Найденные файлы копируются под именем tnsnames[текущая дата].ora.bak (резервные копии), а в оригинальный tnsnames.ora вносятся данные из [\\alpha\symphony\update\config\tnsnames-sym-1000.ora](file:///\\alpha\symphony\update\config\tnsnames-sym-1000.ora)
   3. Отображает окно прогресса обновления tnsnames.ora
   4. По завершении поиска и обновления, процедура должна отобразить окно, уведомляющая о завершении процедуры обновления.
2. Разворачивается новая версия КИС «Симфония» в корневом каталоге [\\betta\Симфония](file:///\\betta\Симфония).
3. Программы [\\alpha\symphony\bin\SymMng.exe](file:///\\alpha\symphony\bin\SymMng.exe) и [\\alpha\symphony\bin64\SymMng.exe](file:///\\alpha\symphony\bin64\SymMng.exe) заменяются на программы, которые тут же закрываются при запуске (можно скопировать notepad.exe под новыми именами или любую другую программу НЕ СИМФОНИЮ)
4. Создаем новый файл [\\alpha\symphony\update\config\sym-1000.scf](file:///\\alpha\symphony\update\config\sym-1000.scf), в котором указана новая версия «sym-1100», новый корневой каталог [\\betta\Симфония](file:///\\betta\Симфония) и новая база данных, указанная в [\\alpha\symphony\update\config\tnsnames-sym-1000.ora](file:///\\alpha\symphony\update\config\tnsnames-sym-1000.ora)
5. Просим пользователей запустить КИС «Симфония» на своих локальных компьютерах.

После запуска КИС «Симфония» на каждой локальной машине с настроечным файлом с версией «sym-1000» будут выполнены следующие шаги:

1. [\\alpha\symphony\SymphonyNext.exe](file:///\\alpha\symphony\SymphonyNext.exe) начнет выполняться с настроечным файлом версии «sym-1000» в режиме Старт с настроечным файлом (п. 6)
2. При обнаружении файла [\\alpha\symphony\update\config\sym-1000.scf](file:///\\alpha\symphony\update\config\sym-1000.scf) программа обновит локальный настроечный файл, вписав в него новую версию «sym-1100», новый корневой каталог [\\betta\Симфония](file:///\\betta\Симфония) и ссылку на новую базу данных, которой пока нет в tnsnames.ora (см. п. 8.2)
3. Программа находит пакет [\\alpha\symphony\bin\scfupdtnsnm.bpl](file:///\\alpha\symphony\bin\scfupdtnsnm.bpl) и запускает его функцию, которая обновляет tnsnames.ora, показывает прогресс выполнения и информирует пользователя по завершении (см. п. 8.3).
4. Если пакет [\\alpha\symphony\bin\scfupdtnsnm.bpl](file:///\\alpha\symphony\bin\scfupdtnsnm.bpl) выполнялся в том же потоке, что и SymphonyNext, то после оповещения пользователя будет выполнен следующий шаг, выполнение [\\betta\Симфония\bin\SymMng.exe](file:///\\betta\Симфония\bin\SymMng.exe) или [\\betta\Симфония\bin64\SymMng.exe](file:///\\betta\Симфония\bin64\SymMng.exe) с новыми настройками.

С точки зрения пользователя, весь процесс обновления будет выглядеть следующим образом:

1. Пользователь запускает настроечный файл.
2. Появляется окно, информирующее об обновлении tnsnames.ora
3. По окончании обновления появляется информационное окно, а после его закрытия
4. Стартует новая Симфония с каталога [\\betta\Симфония\](file:///\\betta\Симфония\bin\SymMng.exe) используя базу данных, развернутую на gamma.

**Необходимость выпуска инструкции администратора (да/нет):** *да*

**Необходимость выпуска инструкции программисту (да/нет):** *да*

**Необходимость выпуска инструкции пользователя (да/нет):** *да*

**Итоги внедрения программы:**

1. Определены стандарты и протоколы запуска КИС «Симфония» с различными конфигурациями.
2. Определены протоколы редактирования конфигураций
3. Определены протоколы обновления конфигураций.
4. Создано приложение, запускающее программу, которая создает главное окно КИС «Симфония».
5. Проект создан на базе библиотеки FireMonkey, что позволит создать единый проект запускающего файла под любую платформу, поддерживаемую средой разработки Delphi.

**Согласовано:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |